MEYERTONS
HOOD
KIVLIN
KOWERT
& GOETZEL
A PROFESSIONAL CORPORATION
PATENTS, TRADEMARKS, COPYRIGHTS & UNFAIR COMPETITION

700 LAVACA, SUITE 800
AUSTIN, TEXAS 78701
TELEPHONE (512) 853-8800
FACSIMILE (512) 853-8801

# FAX

| | |
|---|---|
| To: Examiner Chen | From: Mark S. Williams |
| Fax: 571-270-2071 | Pages: 3 (incl. cover) |
| Phone: 571-270-1071 | Date: August 24, 2009 |
| Re: SN 10/772,518<br>(Attorney Docket No. 5150-38605) | Phone: 512/853-8825 |

● Comments:

Attached please find the Agenda for a Proposed Telephone Interview.

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

| | | | |
|---|---|---|---|
| Application. No: | 10/772,518 § | Examiner: | Chen, Qing |
| Filed: | February 5, 2004 § | Group/Art Unit: | 2191 |
| Inventor(s): | § | Atty. Dkt. No: | 5150-38605 |
| Robert E. Dye, Darshan Shah, Steve Rogers, Greg Richardson, Dean A. Luick | § | | |
| Title: | GRAPHICAL PROGRAMMING SYSTEM WITH BLOCK DIAGRAM EXECUTION AND DISTRIBUTED USER INTERFACE DISPLAY § | | |

**PROPOSED AGENDA FOR TELEPHONE INTERVIEW**

Dear Sir:

This paper is submitted as a proposed agenda for a Telephone Interview scheduled for Tuesday, August 25, 2009. Applicant would like to discuss:

1. The meanings of the distinct (independent) claim elements and their relationships, and what is and is not taught by the cited art regarding these elements. For example, the Examiner's definition of graphical program contradicts the definition as provided in the Specification, which is improper, and so, as a consequence, Huntsman cannot be relied upon to teach sending information regarding a graphical program. Furthermore, and more particularly, Hunstman cannot be relied upon to teach the specific limitation of sending information regarding *the block diagram* of the graphical program, especially since Huntsman's sent information is particular to the user interface, and is relied on to teach this feature.

Here is the definition of graphical program as found in the Specification in p.20, lines 1-9:

...the developer creates or edits the executable portion of the graphical program, which may referred to as a block diagram. A graphical program may include a block diagram comprising objects referred to herein as "nodes" which are connected together to model the program execution logic, data flow and/or control flow. A block diagram node may be displayed as an icon representing the type or functionality of the node. Figure 6 illustrates a simple block diagram. As a developer adds objects to the user interface panel, the graphical programming environment may automatically create a corresponding

object on the block diagram. Such block diagram nodes which correspond to user interface panel objects are referred to herein as user interface nodes or terminals.

P. 4, lines 1-22 provides further definition of "graphical program":

Therefore, Kodosky et al teaches a graphical programming environment wherein a user places or manipulates icons in a block diagram using a block diagram editor to create a graphical "program." A graphical program for controlling or modeling devices, such as instruments, processes or industrial automation hardware, is referred to as a virtual instrument (VI). In creating a virtual instrument, a user may create a front panel or user interface panel. The front panel includes various front panel objects, such as controls or indicators, that represent or display the respective input and output that will be used by the graphical program or VI, and may include other icons which represent devices being controlled. When the controls and indicators are created in the front panel, corresponding icons or terminals may be automatically created in the block diagram by the block diagram editor. Alternatively, the user can place terminal icons or input/output blocks in the block diagram which may cause the display of corresponding front panel objects in the front panel, either at edit time or at run time.

During creation of the graphical program, the user selects various functions that accomplish his desired result and connects the function icons together. For example, the functions may be connected in a data flow and/or control flow format. The functions may be connected between the terminals of the respective controls and indicators. For example, the user may create or assemble a data flow program, referred to as a block diagram, representing the graphical data flow which accomplishes his desired function. The assembled graphical program may then be compiled or interpreted to produce machine language that accomplishes the desired method or process as shown in the block diagram.

Respectfully submitted,

Mark S. Williams, Reg. #50,658
AGENT FOR APPLICANT(S)

Date: 2009-08-24 JCH/MSW